CLAIMS

What is claimed:

1. A method of manipulating data elements in transposing an array of m rows, each row comprising a plurality of n data elements; the transposition is done along the main diagonal down of the matrix. The method has the following steps:

   - Load the contents of row $R(i)$ of the original matrix into the diagonal up $DH(i)$ of a temporary matrix. Where $i = 0$ to $m - 1$, m is the number of rows in the original matrix.
   - Rotate the contents of every row of the temporary matrix to the right by the value of its row index.
   - Store the contents of $DL(i)$ of the temporary matrix into the row $R(m - i$ MOD $m)$ of the original matrix. Where $i = 0$ to $m - 1$

2. The method in claim **1** is modified as follows to perform matrix transpose along the main diagonal down of the matrix. The method has the following steps:

   - Load the contents of row $R(i)$ of the original matrix into the diagonal down $DL(m - i - 1)$ of a temporary matrix. Where $i = 0$ to m, m is the number of rows in the original matrix.
   - Rotate the contents of every row of the temporary matrix to the left by the value of $(i + 1)\text{MOD}n$.
   - Store the contents of $DH(i)$ of the temporary matrix into the row $R((i + 1)$ MOD $m)$ of the original matrix. Where $i = 0$ to $m - 1$
   - The original matrix is transposed.

3. A method of manipulating data elements in transposing an array of m rows, each row comprising a plurality of n data elements; the transposition is done along the main diagonal down of the matrix. The method has the following steps:

   - Rotate the contents of diagonals up $DH(i)$ of the original matrix to the right by the value of their index i. Where $i = 0$ to $m - 1$
   - Rotate the contents of every row $R(i)$ of the matrix resulted from previous step to the left by the value $(2m - 2i)$ MOD m. Where $i = 0$ to $m - 1$.

- In the matrix resulted from previous step, swap the row $R(i)$ with the row $R(m-i-1)$. Where $i = 1$ to $\lfloor \frac{m-1}{2} \rfloor$

4. In the method of claims **1,2,** and **3**, the data elements may be a word of size 8-bit, 16-bit, 32-bit, 64-bit, 128-bit, or larger in a SIMD computer.

5. In the method of claims **1,2,** and **3**, the data elements may be blocks of memory in mesh-connected multi-processors, or any multi-processors that have two-dimensional array configuration.

6. In the method of claims **1,2,** and **3**, the data elements may be blocks of memory cells in a memory array.

7. Methods described in claims **1** and **2** can be used together back to back in a pipelined fashion to overlap steps and save execution cycles, when transposing a set of matrices, as follows:

   - Method of claim **1** starts a transpose by loading DH diagonals up, Rotate, and then Store DL diagonals down.
   - Method of claim **2** is used while method of claim **1** is still storing data. Since both methods of claims **1** and **2** use same DL diagonals in store and load state respectively, stages of load and store of different methods can process data concurrently.
   - Method of claim **2** starts loading data into the DL diagonal immediately after method of claim **1** stores data from the same DL diagonal.
   - Method of claim **2**, then processes the rotation stage.
   - While Method of claim 2 is storing data using DH diagonals, method of claim **1** starts loading data into DH diagonals in the same manner described in the pervious item.
   - Repeat.

8. Method of claim **7** is modified to use method of claim **2** for first transpose, then use method of claim **1** to overlap and repeat as described in claim **7**.

9. A set of registers that are mapped to the same two-dimensional memory array in a SIMD computer that the row registers have access to. This mapping is done according to the following mapping functions:

$$DL(i,j) = R((i+j) \text{ MOD } m,\ j)$$
$$DH(i,j) = R((m+i-j) \text{ MOD } m,\ j)$$

m: number of rows
i: row index 0 to $m-1$
j: column index 0 to $n-1$
R: two-dimensional array with row access

10. The claim **9** allows different sets of registers to share and access a two-dimensional memory array in a SIMD computer using row access pattern, diagonal up access pattern, or diagonal down access pattern.